

예제로 쉬게 배우는

아두이노

김진환 · 장성용 지음

4.1 LED 점멸

4.2 LED 밝기 조절

4.3 RGB LED로 색상 표현하기

4.4 FND 제어

4.5 4-digit FND 제어

4.6 Dot matrix 제어

4.7 Dot matrix를 이용한 애니메이션

LED (Light Emitting Diode)

- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용

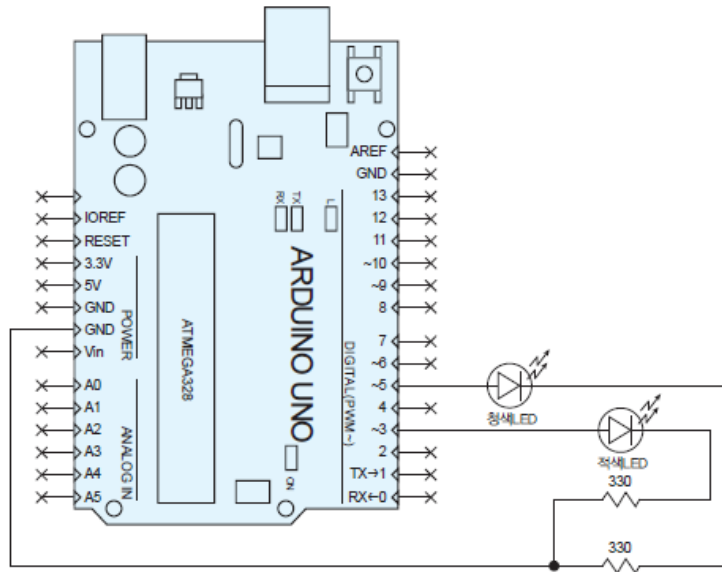


4.1 입출력 핀을 이용하여 LCD 모듈에 표시하기

EX 4.1 변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/3)

실습목표 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

- Hardware**
1. 청색과 적색 LED의 Anode핀을 Arduino의 3번 5번 핀에 연결한다.
 2. Cathode핀에 330 Ω 저항을 연결하여 저항의 반대쪽은 Arduino의 GND에 연결한다.
 3. LED가 연결된 핀에 HIGH신호가 출력될 때 LED가 점등된다.



4.1 입출력 핀을 이용하여 LCD 모듈에 표시하기

EX 4.1

변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/3)

Commands

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT_PULLUP'을 설정한다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. '핀번호'에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW' 를 설정하여 High 혹은 Low 출력을 한다.

Sketch 구성

1. LED의 핀 번호를 설정한다.
2. setup()에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.
3. loop()에서는 하나의 LED를 켜 후 일정시간이 지난 후에 소등하고, 다른 LED를 켜다.

4.1 입출력 핀을 이용하여 LCD 모듈에 표시하기

EX 4.1 변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/3)

실습 결과 LED A와 B가 0.1초 단위로 교차하며 점멸한다.

응용 문제 점멸 주기가 0.1초부터 5초로 0.1초 단위로 증가하였다가 다시 반대로 5초부터 0.1초까지 감소하는 동작을 반복하는 스케치를 작성해 보자.
(hint: delay 명령어의 괄호안의 숫자를 증감시킨다.)

4.2 LED 밝기 조절

밝기 조절 : 디밍 (Dimming)

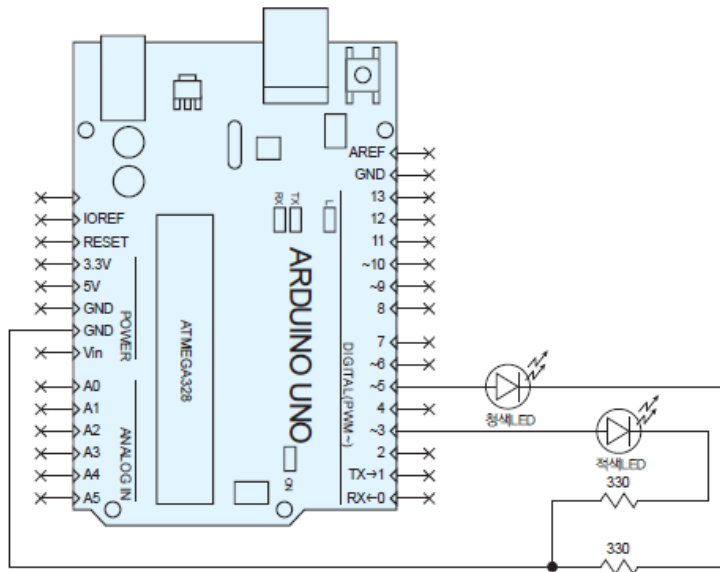
- ✓ LED에 입력되는 전력은 PWM(Pulse Width Modulation)을 이용하여 조절.
- ✓ PWM : 고속의 스위칭으로 High와 Low 신호의 비율을 조절하여 LED의 밝기, 모터의 회전 등을 조절하는 방법
- ✓ Arduino에서는 `analogWrite()` 명령어로 구현
- ✓ Arduino UNO의 경우 3, 5, 6, 9, 10, 11 번 핀이 PWM을 지원한다.

4.2 LED 밝기 조절

EX 4.2 LED 밝기 조절 (1/2)

- 실습목표**
1. 두 개의 LED의 밝기를 조절하자.
 2. 각각의 LED가 교차하며 밝아졌다 어두워 졌다를 반복하도록 하자.

- Hardware**
1. 청색과 적색 LED의 Anode핀을 Arduino의 3번 5번 핀에 연결한다.
 2. Cathode핀에 330 Ω 저항을 연결하여 저항의 반대쪽은 Arduino의 GND에 연결한다.
 3. LED가 연결된 핀에 HIGH신호가 출력될 때 LED가 점등된다.



4.2 LED 밝기 조절

EX 4.2 LED 밝기 조절 (2/2)

Commands • analogWrite(핀번호, 값)
정해진 핀에 아날로그 출력을 한다. '값' 에는 0~255의 값을 넣는다.

Sketch 구성

1. LED의 핀 번호를 설정한다.
2. setup()에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.
3. 밝기를 저장할 변수를 설정한다.
4. 하나의 LED가 밝아질 때 다른 LED는 어두워져야 하므로 이를 조절할 변수를 설정한다.
5. loop()에서는 밝기와 밝기 변수 증감을 위한 변수를 조절하여 두 개의 LED를 교차 점멸시키는 동작을 반복한다.

실습 결과 LED A와 B가 밝기가 변화하며 점멸한다.

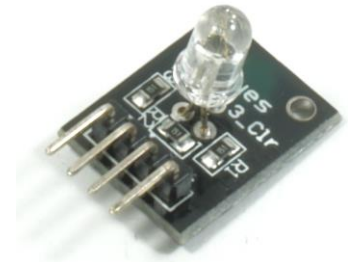
응용 문제

1. 다섯개의 LED를 Arduino에 연결한다.
2. 다섯개의 LED가 순서대로 디밍하는 스케치를 작성해보자.

4.3 RGB LED로 색상 표현하기

RGB LED

- ✓ 빛의 삼원색인 빨강(Red), 초록(Green), 파랑(Blue)빛을 조절하여 다양한 색을 표현하는 LED.
- ✓ 각각의 색이 0~255단계로 조절됨.
- ✓ 간판, 조명기구 등에 사용
- ✓ 모든 색이 출력될 때 백색 빛을 출력

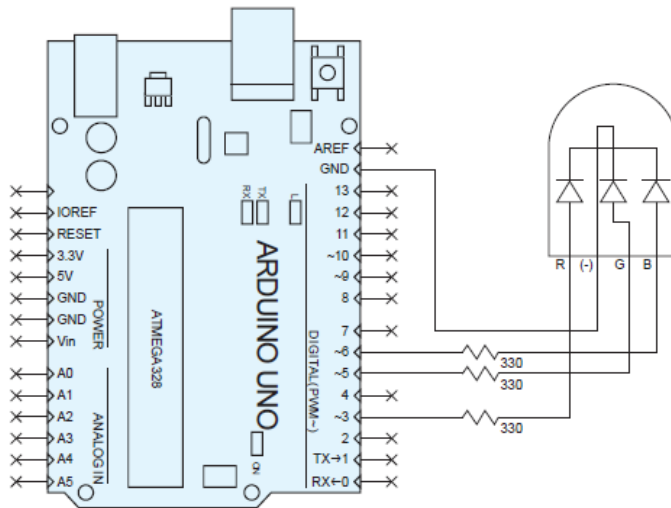


4.3 RGB LED로 색상 표현하기

EX 4.3 RGB LED로 색상 표현하기 (1/2)

실습목표 RGB LED를 이용하여 다양한 색을 표현해 보자.

- Hardware**
1. RGB LED는 Red, Green, Blue의 세 개의 Anode 핀과 공통으로 연결된 캐소드핀으로 구성되어 있다.
 2. RGB LED 단독으로 연결하려면 각 Anode 핀에 330Ω의 저항을 연결해야 한다.
 3. 저항이 내장된 RGB LED 모듈을 사용한다면 별도의 저항이 필요 없다.
 4. Red, Green, Blue의 세 개의 Anode 핀을 Arduino의 3, 5, 6 번핀에 연결한다.



4.3 RGB LED로 색상 표현하기

EX 4.3 RGB LED로 색상 표현하기 (2/2)

Commands

- analogWrite(핀번호, 값)

정해진 핀에 아날로그 출력을 한다. '값' 에는 0~255의 값을 넣는다.

- delay(지연시간)

지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 1/1000초 단위로 넣는다.

즉 1초를 지연시키기 위해선 1000의 값을 입력시킨다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){}

변수의 시작 값부터 조건이 만족하는 경우 '{ }' 내의 명령을 수행한다. '변수의 증분'에서는 1회 명령이 수행될 때마다 변수를 증가 혹은 감소시킨다.

Sketch 구성

1. LED의 핀번호를 설정한다.
2. setup()에서는 LED 출력으로 사용할 핀을 출력 핀으로 설정한다.
3. ledOutput(int Red, int Green, int Blue)라는 함수를 만든다. 적색, 녹색, 청색 LED의 빛의 세기를 조합하여 원하는 색을 출력하는 함수이다.
4. 적색, 녹색, 청색 LED의 세기를 조절하면서 LED에 빛을 출력시킨다.

실습 결과

LED의 색상이 변화하면서 점등된다.

응용 문제

1. 인터넷에서 RGB 색상표를 확인해 보자.
2. 원하는 색상을 출력하는 스케치를 작성해보자.

4.4 FND 제어

FND (Flexible Numeric Display)

- ✓ LED의 조합으로 숫자를 표시하는 장치
- ✓ 7개의 LED를 사용하기 때문에 7-segment 라고도 함.
- ✓ 숫자뿐만 아니라 간단한 기호나 16진수 까지 표현 가능

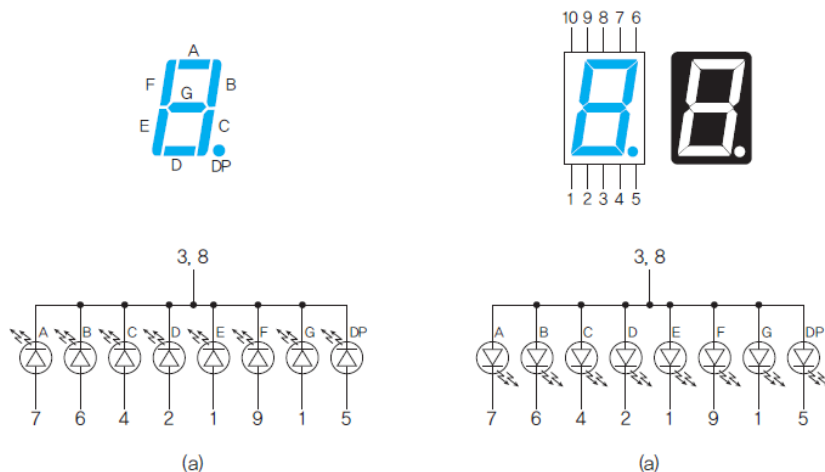


그림 4.4 Common Cathode 형(a)와 Common Anode 형(b)

표 4.1 Common Cathode FND 표시

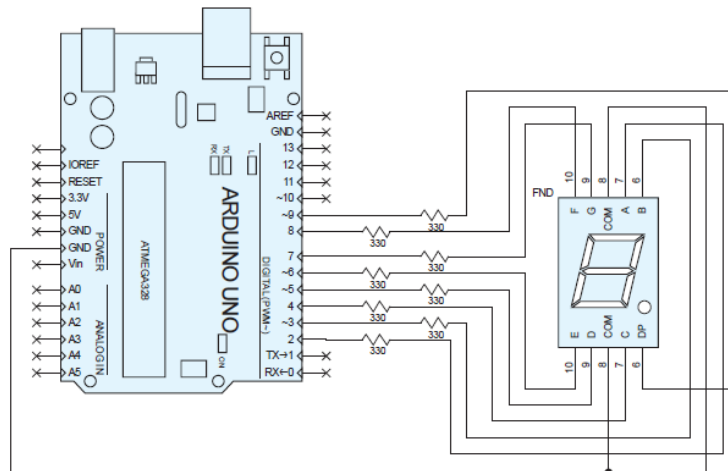
캐소드 공통 7-세그먼트 한 자리 제어 방법										7-Seg. 출력 내용
I/O 포트 출력 내용										
Q0	DP	G	F	E	D	C	B	A	16진수	
1	X	X	X	X	X	X	X	X	X	8. (소등)
0	0	0	1	1	1	1	1	1	0x3f	8. (0)
0	0	0	0	0	0	1	1	0	0x06	8. (1)
0	0	1	0	1	1	0	1	1	0x5b	8. (2)
0	0	1	0	0	1	1	1	1	0x4f	8. (3)
0	0	1	1	0	0	1	1	0	0x66	8. (4)
0	0	1	1	0	1	1	0	1	0x6d	8. (5)
0	0	1	1	1	1	1	0	1	0x7d	8. (6)
0	0	0	0	0	0	1	1	1	0x27	8. (7)
0	0	1	1	1	1	1	1	1	0x7f	8. (8)
0	0	1	1	0	1	1	1	1	0x6f	8. (9)
0	0	1	1	1	0	1	1	1	0x77	8. (A)
0	0	1	1	1	1	1	0	0	0x7c	8. (b)
0	0	0	1	1	1	0	0	1	0x39	8. (c)
0	0	1	0	1	1	1	1	0	0x5e	8. (d)
0	0	1	1	1	1	0	0	1	0x79	8. (e)
0	0	1	1	1	0	0	0	1	0x71	8. (f)
0	1	0	0	0	0	0	0	0	0x80	8. (.)

4.4 FND 제어

EX 4.4 FND 제어 (1/3)

실습목표 Common Cathode FND를 이용하여 0~9의 숫자를 표시해보자.

- Hardware**
1. Common Cathode형 FND는 그림 4.2의 (a)와 같이 3번과 8번핀이 Cathode 핀으로 함께 연결되어 있다. 즉 FND의 3번과 8번핀을 GND에 연결하고 나머지 핀들에 HIGH신호를 주어 FND에 숫자를 표시한다.
 2. GND에 연결되는 3번과 8번핀을 제외한 나머지 핀들에는 FND 내의 LED의 전류를 제한하기 위해 330Ω 저항을 연결한다.
 3. 원하는 숫자를 표시하기 위해선 2~9번핀에 표 4.1을 참고하여 신호를 출력한다.



4.4 FND 제어

EX 4.4 FND 제어 (2/3)

Commands

- void 함수(변수1, 변수2, ...){
};

‘함수(변수1, 변수2)’를 이용하여 { } 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 적는다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’를 설정하여 High 혹은 Low 출력을 한다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){ }

변수의 시작 값부터 조건이 만족하는 경우 { } 내의 명령을 수행한다. ‘변수의 증분’에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

4.4 FND 제어

EX 4.4 FND 제어 (3/3)

- Sketch 구성**
1. FND에 숫자를 표시할 때 어떤 LED를 켤지에 대한 정보를 담은 상수를 설정한다.
 2. FND 동작에 필요한 핀을 출력으로 설정한다.
 3. FND를 동작시키는 'fndDisplay(int displayValue)' 라는 함수를 만든다.
 4. 함수를 이용하여 1초 간격으로 FND에 숫자를 표시한다.

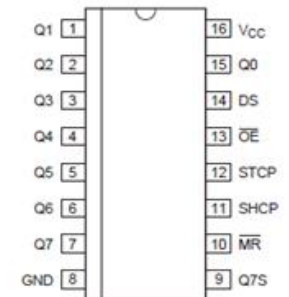
실습 결과 FND의 숫자가 0~9까지 약 1초 간격으로 변화한다.

응용 문제 위의 예제를 0~F까지의 16진수를 표시하도록 스케치를 수정하여 보자.
(hint: LED 표시를 위한 상수에 A~F를 추가시켜서 불러와 사용하자)

4.4 FND 제어

74595 IC

- ✓ 직렬 신호로 입력된 데이터를 병렬 신호로 변환
- ✓ FND의 8개의 LED를 켜기위한 신호를 3개의 신호선으로 입력받아 8개의 FND 신호로 출력
- ✓ shiftout() 명령어로 구현
- ✓ 앞의 예제와 회로도와 스케치를 비교해 보자.

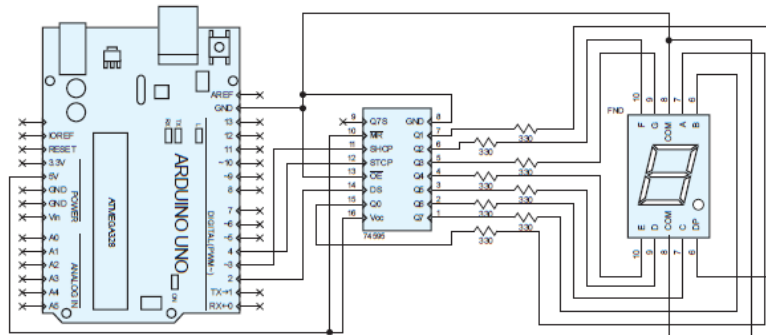


4.4 FND 제어

EX 4.4.2 74595를 이용한 FND 제어 (1/3)

실습목표 Common Cathode FND를 이용하여 0~9의 숫자를 표시해보자.

- Hardware**
1. 예제 4.4.1과 동일한 동작을 하지만 Arduino의 입출력 핀을 절약하기 위해 74595 IC를 중간에 연결한다.
 2. Arduino에서는 2, 3, 4 세 개의 핀을 이용하여 74595 IC로 신호를 출력한다.
각 핀을 Arduino에 연결한다.
 3. 74595 IC의 (MR) $\bar{1}$ 핀과 Vcc 핀에는 5V를 연결하고 (OE) $\bar{1}$ 와 GND핀은 Arduino의 GND에 연결한다.
 4. 74595 IC에서는 DS, SHCP, STCP 핀으로부터 입력된 신호를 이용하여 Q0~Q7 핀에 신호를 출력한다. Q0~Q7 핀을 FND의 Anode 핀에 연결한다.
 5. FND의 Cathode 핀인 3번과 8번핀은 GND에 연결한다.



4.4 FND 제어

EX 4.4.2 74595를 이용한 FND 제어 (2/3)

Commands

- void 함수(변수1, 변수2, ...){
};

‘함수(변수1, 변수2)’를 이용하여 { } 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 적는다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’를 설정하여 High 혹은 Low 출력을 한다.

- shiftOut(데이터 핀, 클럭 핀, 출력비트 순서, 출력 값)

데이터 핀으로는 비트단위로 출력될 핀 번호를 써준다. 클럭 핀에는 데이터가 출력될 때 토글되는 클럭 출력에 사용할 핀 번호를 써준다. 출력비트 순서는 비트 데이터의 맨 왼쪽부터 순차적으로 출력하고자 하면 ‘MSBFIRST’, 맨 오른쪽부터 순차적으로 출력하고자 하면 ‘LSBFIRST’를 써 분다. 출력 값에는 실제 출력할 데이터를 써 준다. 이 때 데이터는 8비트 즉 2진수 8자리의 숫자를 갖는다.

4.4 FND 제어

EX 4.4.2 74595를 이용한 FND 제어 (3/3)

- Sketch 구성**
1. FND에 숫자를 표시할 때 어떤 LED를 켤지에 대한 정보를 담은 상수를 설정한다.
 2. FND동작에 필요한 핀을 출력으로 설정한다.
 3. FND를 동작시키는 'fndDisplay74595(int displayValue)' 라는 함수를 만든다.
 4. 'fndDisplay74595(int displayValue)'에는 'shiftOut()' 명령어를 이용한 FND 동작 스케치를 넣는다.
 5. 함수를 이용하여 1초 간격으로 FND에 숫자를 표시한다.

실습 결과 FND의 숫자가 0~9까지 약 1초 간격으로 변화한다.

응용 문제 위의 예제를 0~F까지의 16진수를 표시하도록 스케치를 수정하여 보자.
(hint: LED 표시를 위한 상수에 A~F를 추가시켜서 불러와 사용하자)

4.5 4-digit FND 제어

4-digit FND

- ✓ FND 네 개를 이용하여 네 자리 숫자를 표시하는 부품
- ✓ Common Cathode형과 Common Anode형
- ✓ FND와 핀 구조는 동일하지만 각 자릿수를 선택하는 핀 추가

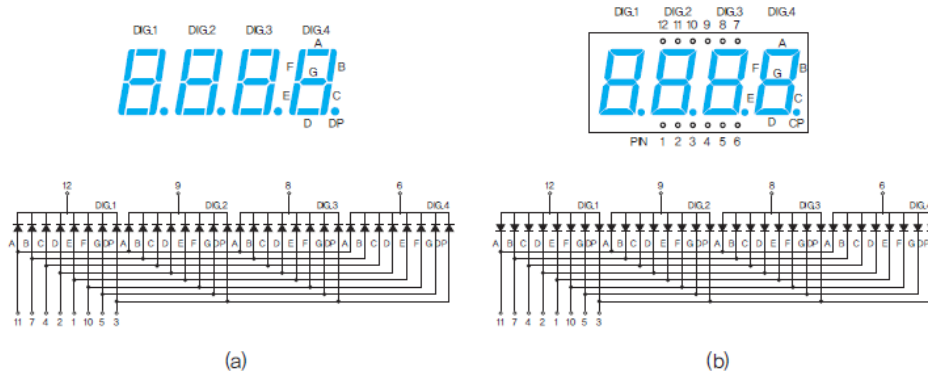


그림 4.6 4-digit FND와 내부 회로, Common Cathode (a), Common Anode (b)



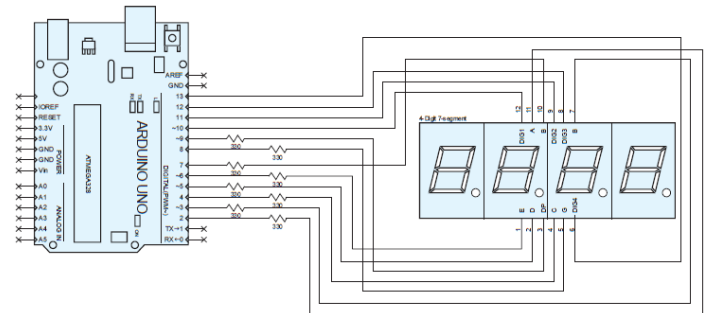
그림 4.7 실험에 사용할 4-digit FND

4.5 4-digit FND 제어

EX 4.5.1 4-digit FND로 0000~9999 숫자 표시하기 (1/3)

실습목표 Common Cathode 4-digit FND를 이용하여 0000~9999까지 1초 간격으로 증가하는 스케치를 작성해 보자.

- Hardware**
1. 4-digit FND는 4개의 FND를 연결한 부품이다.
 2. 각각의 FND에는 DIG1~DIG4 네 개의 핀이 각각의 FND의 Common Cathode로 연결되어 있다.
 3. A~G, DP핀은 하나의 FND를 동작시킬 때와 같이 330 Ω 저항을 통하여 Arduino 2~9번핀에 연결한다.
 4. 맨 왼쪽 FND를 동작시키려면 DIG1에만 LOW신호를 준 상태에서 A~G, DP 핀에 원하는 숫자를 쓰기위한 신호를 주어야 한다.
 5. 두번째 FND를 동작시키려면 DIG2에만 LOW신호를 준 상태에서 A~G, DP 핀에 원하는 숫자를 쓰기위한 신호를 주어야 한다.
 6. DIG1~DIG4에 모두 LOW신호를 주면 모두 같은 숫자가 표시된다.



4.5 4-digit FND 제어

EX 4.5.1 4-digit FND로 0000~9999 숫자 표시하기 (2/3)

Commands

- void 함수(변수1, 변수2, ...){
};

‘함수(변수1, 변수2)’를 이용하여 { } 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 적는다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’를 설정하여 High 혹은 Low 출력을 한다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){ }

변수의 시작 값부터 조건이 만족하는 경우 { } 내의 명령을 수행한다. ‘변수의 증분’에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

4.5 4-digit FND 제어

EX 4.5.1 4-digit FND로 0000~9999 숫자 표시하기 (3/3)

- Sketch 구성**
1. FND에 숫자를 표시할 때 어떤 LED를 점등할 지에 대한 정보를 담은 상수를 설정한다.
 2. FND동작에 필요한 핀을 출력으로 설정한다.
 3. DIG에 연결된 핀을 모두 LOW로 설정하여 모든 FND가 켜지도록 한다.
 4. 예제 4.4에서 설정한 'fndDisplay(int displayValue)' 함수를 응용하여 1초 간격으로 0~9까지의 숫자를 모든 FND에 표시한다.

실습 결과 4개의 FND의 숫자가 0~9까지 1111 단위로 약 1초 간격으로 변화한다.

응용 문제 'XXX1', 'XX2X', 'X3XX', '4XXX' 의 표시가 1초 간격으로 반복하는 스케치를 작성해 보자.
(X:는 꺼짐을 나타낸다)
(hint: DIG1~4에 연결된 핀을 제어해보자.)

4.5 4-digit FND 제어

EX 4.5.2 4-digit FND에 1초마다 증가하는 0~9999 숫자 표시하기 (1/3)

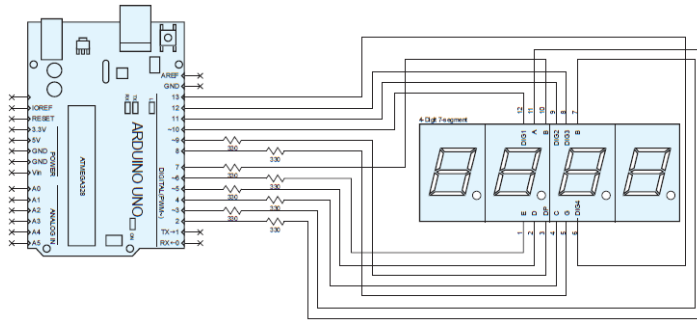
실습 목표 Common Cathode 4-digit FND를 이용하여 0~9999까지 1초 간격으로 증가하는 스케치를 작성해 보자.

- Hardware**
1. 4-digit FND는 4개의 FND를 연결한 부품이다.
 2. 각각의 FND에는 DIG1~DIG4 네 개의 핀이 각각의 FND의 Common Cathode로 연결되어 있다.
 3. A~G, DP핀은 하나의 FND를 동작시킬 때와 같이 330 Ω 저항을 통하여 Arduino 2~9번 핀에 연결한다.
 4. 맨 왼쪽 FND를 동작시키려면 DIG1에만 LOW신호를 준 상태에서 A~G, DP 핀에 원하는 숫자를 쓰기위한 신호를 주어야 한다.
 5. 두번째 FND를 동작시키려면 DIG2에만 LOW신호를 준 상태에서 A~G, DP 핀에 원하는 숫자를 쓰기위한 신호를 주어야 한다.
 6. DIG3, DIG4에 대해서도 동작을 반복한다.
 7. 각각의 FND를 선택하여 점등하는 동작을 빠른속도로 반복하면 마치 모든 FND가 점등된 것으로 인식된다.

4.5 4-digit FND 제어

EX 4.5.2 4-digit FND에 1초마다 증가하는 0~9999 숫자 표시하기 (2/3)

Hardware



Commands • void 함수(변수1, 변수2, ...){
};

‘함수(변수1, 변수2)’ 를 이용하여 { } 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

• pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’ 에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 적는다.

4.5 4-digit FND 제어

EX 4.5.2 4-digit FND에 1초마다 증가하는 0~9999 숫자 표시하기 (3/3)

Commands

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’ 에는 출력하고자 하는 핀의 번호를, ‘값’ 에는 ‘HIGH’ 혹은 ‘LOW’ 를 설정하여 High 혹은 Low 출력을 한다.

- millis()

현재 스케치가 시작된 이후로 경과된 시간 값을 가져온다. 밀리세컨즈(1/1000초) 단위의 값을 갖는다.

Sketch 구성

1. FND에 숫자를 표시할 때 어떤 LED를 켤지에 대한 정보를 담은 상수를 설정한다.
2. FND동작에 필요한 핀을 출력으로 설정한다.
3. DIG1~4중 하나만 점등 한 뒤 해당 DIG 핀에 연결된 자릿수의 표시를 예제 4.4에서 설정한 ‘fndDisplay(int displayValue)’ 함수를 응용하여 표시한다.
5. DIG1->DIG2->DIG3->DIG4 순서로 돌아가며 점등시킨다. 해당 DIG핀에 신호를 LOW 했을 때 해당 자릿수가 점등된다.
6. 빠른시간으로 4개의 DIG핀을 제어하면 시각적으로 모든 FND가 점등된 것 처럼 보인다.

실습 결과

4개의 FND의 숫자가 0~9999까지 1단위로 약 1초 간격으로 변화한다.

응용 문제

숫자가 증가하는 간격을 0.5초로 변경하여라.

4.6 Dot matrix 제어

8 X 8 Dot matrix

- ✓ 여러 개의 LED가 배열되어 문자나 기호를 표시하는 장치
- ✓ 8X8 Dot matrix는 64개의 LED를 이용
- ✓ LED를 빠르게 교차 출력하여 동시에 모든 LED가 제어되는 듯한 착시를 이용



그림 4.8 실험에 사용할 도트매트릭스

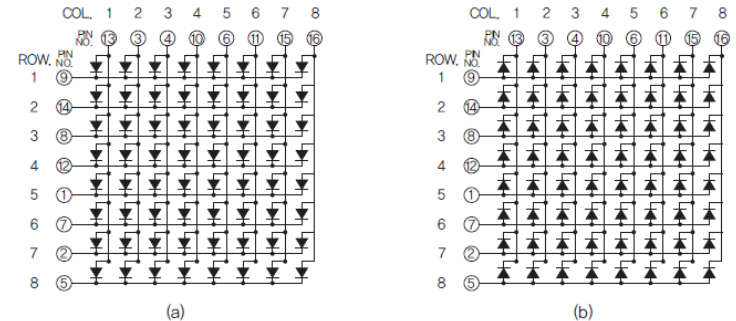


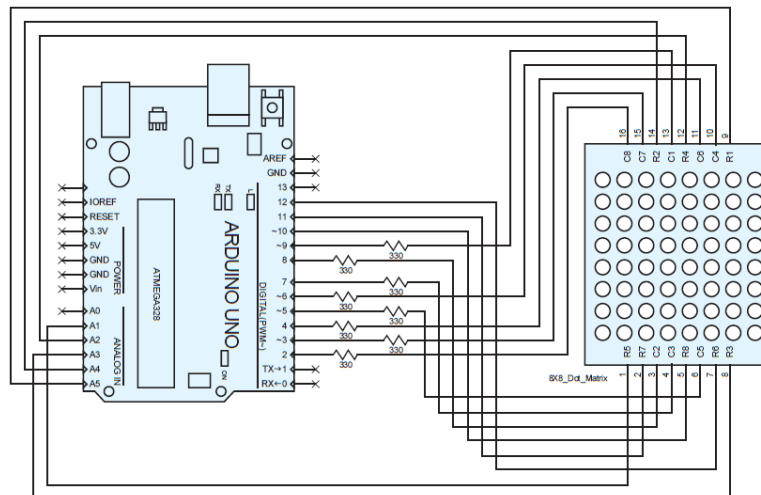
그림 4.9 행에 Anode(+연결)를 연결하고 열에 Cathode(-연결)한 형태(a)와 행에 Cathode(-연결)를 연결하고 열에 Anode(+연결)한 형태(b).

4.6 Dot matrix 제어

EX 4.6 Dot matrix 제어 (1/3)

실습목표 8x8 Dot matrix로 변화하는 막대그래프를 표현해 보자.

- Hardware**
1. 행은 2~9번핀에 연결하고 열은 10, 11, 12, A1~A5 번핀에 연결한다. 행을 연결할 때는 330 Ω 저항을 함께 연결한다.
 2. 실험에 사용할 8X8 Dot matrix는 행(column)에 Cathode, 열(row)에 Anode가 연결된 형태이다. 즉 행에 LOW신호, 열에 HIGH신호를 주어야 Dot LED가 켜진다.
 3. 특정 부분의 Dot LED를 점등하려면 그 부분의 행에 LOW신호, 열에 HIGH신호를 준다.



4.6 Dot matrix 제어

EX 4.6 Dot matrix 제어 (2/3)

Commands

- void 함수(변수1, 변수2, ...){
};

‘함수(변수1, 변수2)’를 이용하여 { } 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 설정한다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’를 설정하여 High 혹은 Low 출력을 한다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){ }

변수의 시작 값부터 조건이 만족하는 경우 { } 내의 명령을 수행한다. ‘변수의 증분’에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

4.6 Dot matrix 제어

EX 4.6 Dot matrix 제어 (3/3)

- Sketch 구성**
1. 8X8 Dot matrix는 그림 4.5의 (b) 그림의 행에 Cathode, 열에 Anode가 연결된 형태를 사용할 것이다.
 2. 행과 열에 출력에 사용할 핀을 모두 출력으로 설정한다.
 3. 점등하고자 하는 행에 LOW 신호를 준 뒤 열에 HIGH 신호를 주어 LED를 점등시킨다.
 4. 행을 하나씩 증가하여 점등시킨다.

실험 결과 C8 부터 C1로 한 칸씩 이동하면서 쌓이는 막대그래프가 출력된다.

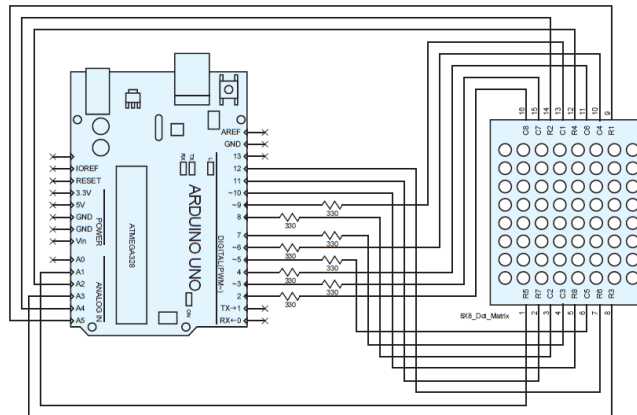
응용 문제 Dot가 한 개씩 이동하는 스케치를 만들어보자.

4.7 Dot matrix 제어

EX 4.7 8x8 Dot matrix를 이용한 애니메이션 (1/3)

실습목표 8X8 Dot matrix를 이용하여 움직이는 화살표를 표시해 보자

- Hardware**
1. 행은 2~9번핀에 연결하고 열은 10, 11, 12, A1~A5 번핀에 연결한다. 행을 연결할 때는 330 Ω 저항을 함께 연결한다.
 2. 실험에 사용할 8X8 Dot matrix는 행(column)에 Cathode, 열(row)에 Anode가 연결된 형태이다. 즉 행에 LOW신호, 열에 HIGH신호를 주어야 Dot LED가 켜진다.
 3. 특정 부분의 Dot LED를 점등하려면 그 부분의 행에 LOW신호, 열에 HIGH신호를 준다.
 4. 1번행만 LOW신호를 주고 1번행에 연결된 열 Dot LED중 점등을 원하는 곳에 HIGH신호를 준다.
 5. 3번의 동작을 1~8번행으로 빠르게 반복하면 마치 모든 행의 Dot LED가 켜진듯이 보인다.



4.7 Dot matrix 제어

EX 4.7 8x8 Dot matrix를 이용한 애니메이션 (2/3)

Commands

- void 함수(변수1, 변수2, ...){
};

‘함수(변수1, 변수2)’를 이용하여 ‘{ }’ 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 설정한다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’를 설정하여 High 혹은 Low 출력을 한다.

- millis()

현재 스케치가 시작된 이후로 경과된 시간 값을 가져온다. 밀리세컨즈(1/1000초) 단위의 값을 갖는다.

- delayMicroseconds()

마이크로 세컨즈(1/1000000 초) 단위로 아두이노의 동작을 지연시킨다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){ }

변수의 시작 값부터 조건이 만족하는 경우 ‘{ }’ 내의 명령을 수행한다. ‘변수의 증분’에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

4.7 Dot matrix 제어

EX 4.7 8x8 Dot matrix를 이용한 애니메이션 (3/3)

- Sketch 구성**
1. 나타낼 그림 데이터를 만든다.
 2. 그림 데이터를 일정시간 표시하는 함수를 만든다.
 3. 시간차를 두고 함수를 호출하여 Dot matrix에 표시한다.

실습 결과 ' >> ' 모양의 화살표가 움직이는 애니메이션이 출력 된다.

- 응용 문제**
1. 화살표의 모양과 방향을 바꿔보자.
 2. 원하는 애니메이션을 출력해보자.

